

**IEPA**

**System Development  
Methodology**

Version update:  
12/20/00

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	1
		Page	1
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Project Authorization/Investigation	Revised	12/20/00

## PROJECT AUTHORIZATION/INVESTIGATION

**Produce Project Authorization/Initialization Document**

**Produce Background & Problem Statement**

**Produce Scope Statement and Project Category**

**Define Hardware Environments** (both production & test)

This phase starts with a notice of project initialization either via the receipt of a Project Initialization Document which has been signed and prioritized by the user area, or via upper management request to start an official Request for Proposal (RFP) to seek contractual assistance in the Development of a Project. In the case where a System is to be developed by an outside consultant the RFP request shall serve as the initial method of project initialization. In the case of an RFP a follow-up Standard Project Initialization Form shall be completed by the Information Services Section for the requesting entity. (See *Section 5.0 Standard Forms/Formats - Notice of Project Initialization* document for the form and procedures on how to complete)

This phase defines the scope of the project. It provides necessary information to determine feasibility of the project and on which platform (hardware and software) and Project Category under which it should be developed.

There are standard hardware platforms, Database Management System (DBMS), application Development tools, programming languages and software products approved for use. Which standard products can be utilized is based up on the nature and scope of the information process.

This phase develops an understanding of the background, and any alternative approaches to the project. It includes a discussion of the resources anticipated to complete such an effort, and roughly estimates the length of time required to implement.

The project is categorized into one of the following four types.

Category I: Major Agency and Bureau system defined as an information process that has significant impact on administration of Agency programs across Bureau/Division or is widely accessed by a combination of headquarters, regional offices, state, local users, internet or intranet access that effects the IRM infrastructure of the Agency.

Category II: Mission critical Bureau/Divisional system that impacts multiple areas within a Bureau/Division and may have to interface with USEPA, our regional offices, the public, internet or Agency intranet and effects the IRM infrastructure of the Agency.

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	1
		Page	2
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Project Authorization/Investigation	Revised	12/20/00
<p>Category III: Localized Information System: An information process that is not a major Agency or Bureau/Division information system but significantly supports accepted program goals and missions and the data needs encompass multiple staff within a Bureau/Divisions office.</p> <p>Category IV: User Owned Information System. Unique, standalone process developed to improve the efficiency or effectiveness of operations for a single user or a small group of users.</p> <p>Agency core software/hardware standards are required to be utilized for all Category I and II information systems. Category III information systems are developed utilizing Agency core software/hardware standards but some functions may encompass approved end user development hardware and software standards. Category IV information systems are developed utilizing approved end user development hardware and software standards which are known to be compatible with our core IT infrastructure.</p> <p>All category I and II systems are to be developed and maintained by the Agency's Information Services Section. Category III Systems that are to be utilizing the Agency's Oracle database management system, the internet/intranet are to be reviewed by the Agency's Information Services Section, and require the sign-off authorization of the Information Services section Database Administrator's, Security Officer and other IS management staff to ensure their integration into the Agency Infrastructure.</p> <p>It is during this phase that a project, or parts of a project, may be determined to be candidate(s) for developing a prototype. Both the user implementation needs (critical deadlines) and automation needs (functional requirements) are evaluated.</p> <p>The analyst assembles process descriptions offered by individual users or at user meetings into a statement of what the whole system or application has to do. Working intimately with users, the analyst identifies major functions, key attributes and which processes create and use them. The analyst organizes these into a summary document and then meets with users, and management to hammer out the Scope. Scoping is the process of deciding which processes will be done, which will not, which can wait, and which are outside the realm of the application being developed.</p> <p>The three items listed above represent deliverables that are to be agreed upon (subject to sign-off).</p> <p>Normal phase processes include:</p> <ul style="list-style-type: none"> <li>- Project authorization document</li> <li>- General data gathering and analysis</li> <li>- Background narrative</li> </ul>			

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	1
		Page	3
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Project Authorization/Investigation	Revised	12/20/00
<div><ul style="list-style-type: none"><li>- Scope narrative</li><li>- Alternatives and expanded options narrative (optional)</li><li>- Estimated cost-benefit (including initial costs vs. ongoing costs) (optional)</li><li>- Resource commitments</li><li>- Rough implementation schedule</li><li>- List of open issues</li></ul></div>			

--

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	1
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00
<p><b>ANALYSIS</b></p> <p><b>Produce Functional Decomposition</b> (actions needed)  <b>Produce Process Model</b> (process descriptions, event/role diagrams)  <b>Produce Data Entity/Attributes descriptions and ER diagram</b>  <b>Produce External Interface Requirements</b>  <b>Produce Data Administration Requirements</b>  <b>Produce Hardware and Software Configuration</b>  <b>Produce Conversion Approach</b>  <b>Produce Design Phase Work Plan</b>  <b>Produce Phase Sign Off</b></p> <p>A problem must be understood before it can be solved. Analysis is the discipline of studying and documenting requirements, so a system or application can be focused on meeting those requirements and performing the supporting job for its users.</p> <p>Analysis is not pure and detached. It must be conscious of the scope that was defined in the Project Authorization/Investigation phase. Although analysts must keep an open mind with regard to design, it is proper to allow external or internal constraints to affect their definition of the requirements, i.e., deadline for implementation, budget or staffing levels, existing regulations or laws affecting choices, highly integrated and related processes, or existing equipment and/or software that must be used or constraints of a purchased package.</p> <p>Data and information gathering interviews or meeting with the users are the mechanisms used in obtaining requirements. Familiarization with any legislation, rules, or policies that govern the business function are also critical in understanding the needs and requirements.</p> <p>This phase defines the functions and processes needed, as well as the data structure that will be used in the project. The data structure should identify all data elements (attributes) to be addressed, including a data conversion plan, if this data structure is to replace any other data structure. The meaning of each element (attribute) should be cited, and indexes identified.</p> <p>During this phase the hardware environment is accessed and generally tested for usability. If not previously existing, hardware is installed and configured per requirements specific to the project. Special attention should be given to aspects of the hardware environment which may take longer to put into place. This will help ensure timely completion of the project. These tasks should be identified and an approach should be devised to pull all activities together in a coordinated manner.</p>			

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	2
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00
<p>One of the earliest factors to be considered in system and application development is the need for names for data processing objects, like entities, key fields and attributes. The assignment of these object names starts as early as the analysis phase. The establishment of naming conventions within a shop for various objects recognizes the fact that every piece of documentation communicates to people later in the life cycle of a system. Developers write for themselves in their next incarnation as maintainers of a system and for their users. Naming conventions are established so that they assist in providing a unique identification for an object within the context in which it is defined, thus providing a reference that is easily identifiable and recognizable. The context of a field attribute is the record or table in which it appears. The context of a program variable is the program or module. The context of a login ID is the network or computer to which it applies. Please refer to "Standard Naming Conventions" Section 1.0 and applicable tool standards for system component naming criteria.</p> <p>Standard Analysis processes include:</p> <ul style="list-style-type: none"> <li>Interview planning (optional)</li> <li>Requirement gathering meetings and interviews</li> <li>Functional diagrams</li> <li>Data flow analysis (optional)</li> <li>Process descriptions</li> <li>Data element (attributes) identification and definitions (start of)</li> <li>Entity identification and definitions</li> <li>Entity-relationship diagrams</li> <li>Start conversion plan formulation (if needed)</li> <li>External interface analysis and definition (if applicable)</li> <li>Packaged software compliance analysis (if applicable)</li> <li>Data Administration</li> <li>Purchased equipment and software (if applicable)</li> </ul> <p>From these processes the following deliverables are produced.</p> <p><u>Functional Decomposition Diagrams</u></p> <p>Functional decomposition is the breakout of major business activities or roles. In a large system these major business activities break into organizational units or individual applications within a major system. For an application, functional decomposition is the breakout and hierarchical representation of actions needed. Decomposition diagrams show how processes are subordinate to one another in a hierarchy.</p>			

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	3
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00
<p>Example of functional decomposition at system level:</p> <p>Permitting</p> <ul style="list-style-type: none"> <li>Permit Process Tracking</li> <li>Permit Fee</li> <li>Permit Application Process <ul style="list-style-type: none"> <li>Permit Issuance</li> <li>Permit Denial</li> </ul> </li> <li>Permit Renewal</li> <li>Permit Reconsideration</li> <li>Permit Filing</li> </ul> <p>Example of functional decomposition at an application level:</p> <p>Permit Processing</p> <ul style="list-style-type: none"> <li>Application Received</li> <li>Application Reviewed <ul style="list-style-type: none"> <li>Conference Meeting</li> <li>Permit Posting</li> <li>Public Notice</li> <li>Public Hearing</li> </ul> </li> </ul> <p>CROPA Coordination</p> <p>Permit Decision</p> <ul style="list-style-type: none"> <li>Permit Denial</li> <li>Permit Issuance</li> </ul> <p>Permit Renewal</p> <p>Functions come in two varieties; those that use some data and those that do not. The ones that don't are assumed to be manual. The process of functional breakdown continues through the analysis and design phases of system and application development.</p> <p>Hierarchical diagramming of functions can suggest dependencies or triggering that is needed, i.e., one function causing another function to start.</p> <p>From this level of functional representation the user management will decide which functions to automate or which functions need to be grouped for automation.</p>			



# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section 2

Chapter 2

Page 4

Section Title System Development Methodology

Issued 2/1/97

Chapter Title Analysis

Revised 12/20/00

## Data Flow Diagrams (optional)

Data flow diagrams show how data flows through a sequence of procedures or processes in a business function. They emphasize who does what, with which data and in what order. They are useful in visualizing, i.e., depicting the application as a whole. If a entirely new business function is involved, a data flow diagram will assist the user in charting the flow of work.

## Process Descriptions

Process descriptions describe how data is used. They tell what they do, which tables and data elements they operate on, and whether they create, update, delete or simply reference those data elements. (See *Process Description sample 2.0* below)

It is in the process descriptions that process rules and requirements are stipulated such that the inference of special editing, interfaces, data references and process invocation by other processes, etc., are defined.

Process descriptions are often referenced as a business transaction that implements a specific business function or process. A business transaction when defined may entail a group of related data entry screens, reports, database updates, inquiries or reports that are needed to implement the business process.

## *2.0 – Two Examples of a Process Description Format for a School:*

### **Process #1:** Send out admission letters

Description: Admissions are done in February of each year. Each applicant will have submitted an application sometime after October 15 of the preceding year.

The Head of School who handles the grade (Nursery, Lower School (K-3) or Upper School (4-6) will follow these steps in admitting children to the school.

Determine the number of openings at the grade level. The Nursery three-year-old class is all new each year. For all other classes the Head of School must estimate attrition by the number of deposits received with re-enrollment contracts, the number of applications out, that is, transcripts requested by students who will rise to the grade level in question. The estimate has to be an educated guess based on experience. The Head of School knows the historical ratio of applications out to acceptance. Students whose families move over the summer and those who leave for public schools typically give no indication of their intention to leave.

--

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	5
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00
<p>Create an ordered admissions list for the class. At all levels preference is given first to children of parishioner, second to siblings of current students, then to children and siblings of alumni, and lastly to families with no association with the school. The process is subjective. The Head of School reviews transcripts and notes of interviews held with parents and applicants in an attempt to form classes which are balanced in terms of gender and ability. Applicants whose transcripts and interviews indicate the school cannot adequately serve them will be sent rejection letters.</p> <p>Weigh the likelihood of each applicant to accept the offer of admission. The estimate is very subjective, depending on the grade level and the past association of the family with the school.</p> <p>Create a cutoff point at which the cumulative weight totals the number of positions open at the grade level. If there are four openings in a class and the names are in the admissions priority shown, the school would send admissions letters to the top six names in the expectation that two will not accept.</p> <p>Send letters of admissions to those who make the cut. Send wait-list letters to others who are qualified, but fall below the cut.</p> <p><u>Data References:</u> This process updates the Role entity to change an applicant's status from Applicant to Admitted. It uses the Student Mailing Process to compose form letters.</p> <p><u>Invoked by:</u> Admissions process commencement date of February 1 each year.</p> <p style="text-align: center;">*****</p> <p><b><u>Process #2:</u></b> Mailing List Creation</p> <p><u>Description:</u> A standard process to assemble mailing lists for form letters. The process accepts as input the role for which the mailing list is to be created, the criteria for choosing names within that role, and additional attributes to select for inclusion in the body of a mailing. Its output is a file of names, addresses and variable text to include in the body.</p> <p><u>Data References:</u> This process selects from the Role, Person, Relationship and Address Tables. It makes use of the fact that by the constraints of the database that every person is either a head-of-household and has an address, or has a member-of-household relationship with one and only one person who is a head-of-household. The rules for finding an address are:</p>			

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section	2
Chapter	2
Page	6

Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00

1. Use the person's summer address if the person is a head of household and it is summer, otherwise,
2. Use the person's primary address if the person is a head of household, otherwise,
3. Locate the single relationship which indicates the head of household in which the person lives. Use that person's address as determined by rules 1 and 2.

Invoked by: Mailings to students, employees, and potential donors.

## Data Element (attribute) descriptions

(see defined Data Element Description Format 2.1 below) Data elements are captured as they are needed in the analysis phase. Documentation must contain a data element name, its format and length, description of how the data element is used within the system, and the description of edits and computations which involve the data element.

### ***2.1 Data Element Description Format of a Typical Attribute***

Name: Social Security Number.

Standard Attribute Name: SSN

Definition: Used as the key field for identifying people. The system generates artificial SSNs for foreign nationals and people who could not be expected to provide their SSN.

Format: 11 characters. Alphanumeric for generated SSNs, leading alpha followed by 10 numeric. May require a separate attribute to indicate the SSN type, real or generated.

Edits: Real SSNs must be 11 characters long, all numeric. Generated SSNs must have a leading letter followed by 10 digits.

Computation: Rules will be written to search appropriate databases and return confirmed duplicate people and suspected duplicate people to prevent their insertion into the database. An Oracle sequence will create the numeric portion of generated SSNs. Processes that access the sequence will be responsible for invoking the standard subroutines that search for potential duplicates. There must be a process to change the key when a real SSN does become available for persons possessing a generated one.

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>	Section	2
	Chapter	2
	Page	7
Section Title      System Development Methodology	Issued	2/1/97
Chapter Title      Analysis	Revised	12/20/00

### Entity Definitions (see Entity Definition Format)

In relational technology entity definition is the center stone of defining a good database structure that will meet the requirements defined by the users. Entity identification usually will translate into relational table definitions. Data elements are assessed and grouped and attributes to the entity. Entity definitions usually contain the following items: (*See Entity Definition standard format 2.2*)

- Entity name, usually in a format that can be used as a table definition.
- Description, telling what the entity is. The description includes, who created it and where it is used. Subtypes should be covered.
- Unique key attributes, those that will serve as a primary key for the table.
- Foreign key attributes (data elements), those that will determine how the entity relates to other entities.
- Volumes if available; number of records anticipated, the frequency of creation, update and deletion and percentage of nulls.
- Subtype indicators, identification of things that act similarly but must be recognized as distinct.
- Special considerations. Unique processing requirements.

### **2.2 Two Entity Definitions of a Probable School Database:**

#### **DEFINITION #1:**

Entity Name: Person

**Description:** A person associated in some way with the school, such as an applicant, alumni, parent, student, trustee, employee, etc. A single person may be associated with the school in multiple roles; see role entity for more detail.

Aliases and name changes are not normalized through the use of multiple records. There is one attribute (data element) for "also known as".

**Unique Identifier:** SSN is primary key

**Volume:** 1000 records, with 20 add, and 50 updates per year. Except during application period of Aug. through Oct. when there are 100 adds per month.

**Special Processing:** Records of terminated employees will be archived by an annual batch process three years after their date of separation. Records of ex-students and

applicants will be kept online for one year.

\*\*\*\*\*

## ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section 2

Chapter 2

Page 8

Section Title System Development Methodology

Issued 2/1/97

Chapter Title Analysis

Revised 12/20/00

### DEFINITION #2:

Entity Name: Role

Description: One role that a person plays within the life of a school. The same person may be Active in many roles, which is why roles cannot be treated as a subtype of person. Subtypes and sub-subtypes of roles are:

- Employee
  - Administration
  - Teaching Staff
  - Custodial
- Parent or Guardian
- Student
- Alumnus
- Trustee
- Benefactor

Unique Identifier: SSN (foreign key) and role code

Volume: Approximately 25% more than person entity, with same transaction volatility.

Special Processing: Employment data elements are carried in the employee subtype.

The entity and attribute data collected is then analyzed and entity/data is normalized is done and/or refined.

### Entity-Relationship Diagram

Entity-relationship (ER) diagrams are the most universal piece of documentation in a relational system. They show how entities are related to one another based on shared attributes (keys). One to one, one to many, many to many relationships (these must be resolved in constructing your data structure for they cannot be implemented) are depicted and how you get from one table to another. The ER diagram, entity and attribute descriptions become the foundation for the database design. In constructing ER diagrams, standard industry constructs and symbolism's are to be utilized. Consult an ER diagramming techniques manual or see your information system support analyst for assistance. (*See Sample Entity Relationship Diagram Convention in Section 6.0*).

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section 2

Chapter 2

Page 9

Section Title System Development Methodology

Issued 2/1/97

Chapter Title Analysis

Revised 12/20/00

## External Interfaces (optional)

This documentation identifies and discusses any external factors or conditions that have to be identified. It identifies other systems that will provide data to the system or use data from it. The topics cover:

- What data elements (attributes) need to be exchanged with other systems?
- What is the data element format in the other system?
- On what schedule is the data available for input, or must be available for output?
- What is the volume?
- On what medium is the data needed? What is the conversion involved, if any?
- Are there authorizations needed for this data?
- Are there confidential elements that require security?
- What other interface or external consideration needs for access, update, or data sharing are needed and should be addressed in designing the system or application.

### ***2.3 Example of an External Interface Description:***

Interface: Payroll Service Payee Information Update

#### Entities/Attributes Required:

The following attributes are required in a fixed length ASCII file in which the format is specified by the payroll service:

Employee Name  
Employee Address  
Monthly Salary  
Number of Dependents  
Payroll Deduction Data: Type and Amount

Volume: 50 changes per pay period, except in June hiring season when it is 200.

Medium: Modem transmission to ABC payroll service.

Timing and Frequency: Transmissions made evening of 12th day of month and 3 days before end of month. PC placed in auto-answer mode to respond to auto-dial by payroll service.



<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	10
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00

#### Packaged Software Compliance Analysis (optional):

If a new application is being considered as a pre packaged solution, it must be looked at and evaluated against its ability to handle the system and user requirements. Generally some form of cross-reference table of requirements and vendor ability to meet them is made. Again other influencing factors must come into play when looking at packages such as; will it run on current hardware, software, is it modifiable, how much customization is required, etc.

#### Data Administration Consideration

Data Administration documentation is a free form narrative that examines the backup, archiving (retention), and disaster recovery needs of the application or system. In most cases a shop's standard backup procedures will handle the majority of application backup needs. On occasion, due to volume, or criticality of data, the need for a special backup, archival or data retention will exist. These needs have to be discussed, analyzed and documented in the analysis of the system or application so they can be put into effect during development.

#### *2.4 Example of a Data Administration Document Using Our School for Pertinent Tables.*

##### **ROLE AND PERSON ENTITIES (tables)**

Backup: The daily backup requirements are the same as for the entire database, that is, daily backup. The backup plan requires that the three most recent days' backups be available for recovery purposes; that the previous month-end backup be kept in off-site storage to protect against destruction.

Archiving: The role entity controls archiving requirements for itself and also for the persons entity. Records for a person can be deleted when person no longer has a role in the life of the school. Each subtype of the role entity has different archiving characteristics.

Applicant subtype records are of no interest once the school year for which they were made is over, thus they can be deleted without archive at the end of each school year.

By state law employment records must remain available for five years after termination. Person and role records of employees will be left on the system for that period, then archived to microfiche. An entry in the finder entity, indexed by SSN and last name, will provide a quick online reference to the archived records.

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	2
		Page	11
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Analysis	Revised	12/20/00

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section

2

Chapter

2

Page

11

Student and parent records are numerous and of little interest once a student has left the school. The role and person records of departing students and their families will be archived to microfiche at the end of each school year, providing those people do not continue in other roles. A row will be added to the finder entity for each person whose records are archived.

## Conversion Approach

A conversion approach is developed if applicable. A conversion approach is to identify all actions that will be needed to convert an old system to a new system and/or a new system back to an old format. Generally this will be in a narrative step format.

## Design Phase Work Plan

A work plan is developed for the next phase and task assignments are made depending on resources to be placed on project.

## Phase Sign-Off

An official sign-off is to take place after the completion of the Analysis Phase. Signoff is obtained from affect IT section and Bur/Div responsible management. This sign off is the guide requirements documents for the next phases of the System/Application development software. SECTION 5.0 Standard Forms contains the `Sign-Off Document`=to be used for these purposes. Sign-off documents are to be kept with the system and application documentation manuals.

Section

2

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Chapter

3

Page

1

Section Title      System Development Methodology

Issued

2/1/97

Chapter Title      Design

Revised

12/20/00

## DESIGN

**Produce Bounded Discrete Modularity** (if iterative or parallel approach is chosen)

Finalize Entity/Data Attribute Description and QA Document

Produce Database Definition

**Produce General User Access Level Assignment**

**Produce Definition of Menu Structure**

**Initiate Global Parameter Definitions**

**Produce Module Specifications**

**Produce Conversion Specifications**

**Produce Development Work Plan**

**Produce Phase Sign Off**

Having a clear understanding of which business functions and processes that are to be supported, the Design Phase outlines the solution. It is here, if needed, where a breakout or prioritization of development sub-components can be made for parallel or staged development and implementation such that functions with imposed implementation deadlines can be accomplished first. This phase produces prototyping of menus and module definitions. The processing requirements are then categorized into transactions that will accomplish each request through queries, table update, aggregation or distribution of various tables as well as the form of the transactions and distribution of processing. Trade-offs among system efficiencies, security, network traffic and response time will be considered. User access will be assigned.

Normal design process includes:

Finalizing/Organizing Entity and Attribute Descriptions

Database definition

Security and access level narrative

Module specifications (reports, screens, batch processes, interphase processes)

Identify common interfaces to other applications or systems

Manual processing descriptions and procedures

Prototyped screens and menus

Walkthru of specification among users and developers

Backup and archiving needs defined

Configuration and installation of hardware and/or software

Phase sign off

Following are defined deliverable from this phase:

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section	2
Chapter	3
Page	2

Section Title      System Development Methodology

Issued              2/1/97

Chapter Title      Design

Revised            12/20/00

## Entity/Data Attribute description and QA document

This document is the completion and formalization of the Entity and Attribute Descriptions started in the Analysis Phase. These definitions are to be consolidated in a document where each Entity Definition standard format criteria is presented followed by each entities attributes and their standard required description (*see Documentation Example in Section 3.0*). It is at this time that the responsible IT and Bureau/Division QA tester/verifier is assigned and indicated in the Entity Description format under Quality Assurance Verification heading:

### **Quality Assurance Verification:**

Original:

Date: \_\_\_\_\_ Responsible: IT Verifier \_\_\_\_\_ BUR/DIV Verifier \_\_\_\_\_

Latest Modification:

Date: \_\_\_\_\_ Responsible: IT Verifier \_\_\_\_\_ BUR/DIV Verifier \_\_\_\_\_

The date is filled into the entry during the Development Phase when testing is complete by both parties.

## Database Definition

The database design establishes the database structure of table, views, and indexes to be used by the application. These are the real world implementation of the entities, attributes and relationships documented in the analysis phase. The following documentation is developed during this process.

- Table definitions, derived from the entity definitions in the analysis phase, table definitions including foreign key constraints.
- Column definitions, derived from the attribute definition in the analysis phase. Attribute lengths and data types established at this time, as well as any validity criteria of column contents (constraints) and any formulas for computing derived columns.

## Database Administration and Security/Access Level Design

Database Administration and security functions are also set up which may include; which user IDs own the tables, how the tables will be distributed into table space, what views will be used for convenience or to restrict access. Criteria such as performance, data volumes, access security, data integrity, backup and general user administration are all factors to be considered in how the aforementioned data requirements are structured.

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	3
		Page	3
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Design	Revised	12/20/00

### Menu Structure

One of the designer's first tasks will be to work out how functions will be presented to the user. This begins by transforming the functional decomposition done in the analysis phase into a more detailed composition. Due to the volatility of menu structures throughout the application development process, it is recommended that they be presented in the design phase in simple word processing outline form, thus easily updateable in subsequent phases.

In designing menus the following considerations are often used to determine their structure.

- Group functions on a menu by their users
- Keep user privileges in mind. Users with different privileges need different menus.
- A given process might appear multiple times in a menu structure for use by different users or simply to make it close at hand wherever a user might need it.
- A menu process organizes functions by the order in which they are used and the frequency with which they are needed. Infrequently used processes might be grouped into low-level menus without regard to how they fit within a functional decomposition.
- It is often convenient to group like functions together, such as updates on one menu and reports on another, to make them easier to find.

The designer should refer to discussions found in their development tool manuals or other reference material for further guidelines or recommendations on menu development. If an application or system already has standards established for menu organization these are to be followed for subsequent system additions or modifications.

In the design phase interaction among processes, below the menu level should also be discussed, i.e., a screen or form can call another form, or a batch process invoked by a menu can execute several batch programs. The discussion should also address batch processes that do not go through the menu structure, such as those that are invoked by a system operator from the command line.

### Global Parameters

Part of application design is to define parameters used by programs to communicate with one another. These include the substitution parameters/global variables used by menus or screen forms to communicate with each other and processes that they invoke. Systems may also use operating system shell variables and temporary files to communicate among programs and processes. These system variables or global parameters need to be defined and documented.

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section	2
Chapter	3
Page	4

Section Title      System Development Methodology

Issued              2/1/97

Chapter Title      Design

Revised            12/20/00

It is important that these be rigorously defined to the extent that the programs that create or use them share a common interpretation of their format and meaning. Below is a suggested format for defining these parameters and variables. The design phase starts the initial list, then in the development phase, the programmers should append or modify the list as needed in the course of writing the system.

<i><b>Substitution Parameter/Global Variable Name</b></i>	<i><b>Description</b></i>	<i><b>Set by</b></i>	<i><b>Used by</b></i>	<i><b>Format</b></i>
&UN	User Name	SQL Menu Logon	All process	CHAR (30)
&RM RPT_MONTH	Current Month for rept. purpose	Operating system shell	All application processes	CHAR (6) Format YYYYMM

## Process/Event Driven Module Specifications

A process is a procedure or module and can be one of any of the following:

- A screen or in Oracle SQL\*Forms, one entire form or a block within a form.
- A menu, one page of the menu or the whole menu system.
- A coded procedure, in Oracle a PL/SQL code block such as a Forms trigger or a stored procedure.
- A report module
- Another module type, such as graphics process.
- A batch program invoked from menu or the operating system in any language.
- A subroutine called by any of the above, such as a user exit.
- An operating system shell script.

In a 4GL. environment many developers feel that they know what the user needs and see no need to write it down in advance, for if they get it wrong they will correct it as they prototype. In many instances where only simple one and two table forms or reports exist, they are correct, because the data implies the process. But even in these situations if some form of specifications are not kept, processing controls or features are not remembered as being required or not incorporated as time goes by. For more complex situations, specifications are indispensable especially in situations, such as:

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>	Section	2
	Chapter	3
	Page	5
Section Title    System Development Methodology	Issued	2/1/97
Chapter Title      Design	Revised	12/20/00

- Involved processing logic within mainline procedures
- Library objects, pieces of code that will be used in several places within the system. The written specifications ensures that the programmer or subsequent user programmer of the routine all agree on the number, format, and sequence of parameters to pass back and forth.
- System-wide procedures, such as error conditions to check and error message presentation.
- Complex program logic, which might evolve decision charts to ensure all cases or a condition are covered, flow charts to handle comparisons or conditional flow logic, computations or error conditions.
- Screen functionality, and overall GUI formatting should also be specified. If standard shop or application system defaults are used this should be stated in the specification, and the exceptions per module noted.

Process module specifications are started at the module design level either by the module designer or the developer. A meeting between the user, designer and developer is held to hammer out initial requirements and to go over system and application wide requirements such as error message presentation, application standard GUI criteria and needed functionality, calculations, verification, etc. In a 4GL. environment the designer/developer are often the same, thus they initiate the module specification via narrative form then prototype the module. Updating of process specifications continues through out the development phase, and all change levels of testing and modification until user acceptance.

This documentation is to serve as the user requirements document and the formalized communication document on the agreed to functionality of the module. This documentation is to be constructed utilizing the Agency's standard word processing package and standard format described below. A directory for each system is to be established and a subdirectory should be associated with each module name.

A Quality Assurance check box entry is to be contained within each module spec such that the responsible Quality Assurance IT and Bureau/Division tester/verifier individuals can be assigned and their names entered. After successful testing of the module (see Development Phase Application Testing), this date box will be completed with the date final quality assurance testing was complete.

At a minimum, a module spec is to contain the following if applicable: *(See Section 5.0 Standard Forms/Formats 5.3 `PROCESS MODULE SPECIFICATION FORMAT=for content definition. For sample specs see 3.1 Documentation Examples.)*



<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>	Section	2
	Chapter	3
	Page	6
Section Title    System Development Methodology	Issued	2/1/97
Chapter Title      Design	Revised	12/20/00

- a) Module Name:
- b) Standard Module Name:
- c) Description of the Module:
- d) Location: (location of data and physical path of where module is located)
- e) Originator:
- f) Inputs:
- g) Toolbar
- h) Referenced Objects:
- i) Database Trigger(s) Fired
- j) Output:
  - Tables/Files:
  - Reports:
    - Report Title:
    - Subtotals:
    - Grouping:
    - Sort Order/Indexing Fields Displayed:
    - Report Layouts:(Samples)
    - Type of Output (Screen/Printer)
- k) Processing Functionality:
- l) QA Process Verification Indication Entry:

### Quality Assurance Verification:

Original:  
           Date \_\_\_\_\_ Responsible: IT Verifier \_\_\_\_\_ BUR/DIV Verifier \_\_\_\_\_  
 Modification:  
           Date \_\_\_\_\_ Responsible: IT Verifier \_\_\_\_\_ BUR/DIV Verifier \_\_\_\_\_

A specification walkthru should be held to iron out the kinks in the logic, if needed. These walkthrus should encompass multiple modules if dependencies are inherent. The end user, designer and developer should attend the walkthru.

Applications are easier to use if screens (forms), menus, and reports follow a standard format. At a minimum, design and design documentation must address the format of heading lines, conventions for drawing boxes around blocks, conventions for pop-up windows, page numbers, field alignment, cursor movement, use of color and use of fonts in printing. These overall application specifications should be documented up-front in a common application specifications section in the systems documentation and distributed to all developers. (*See Section 2.0 Screen/FORMS Design Standards: `Oracle Application FORMS Standards and Guidelines=or `FoxPro Application Screen Standards and Guidelines`*)

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section	2
---------	---

Chapter	3
---------	---

Page	7
------	---

Section Title	System Development Methodology
---------------	--------------------------------

Issued	2/1/97
--------	--------

Chapter Title	Design
---------------	--------

Revised	12/20/00
---------	----------

## Conversion Specification Document

If in the Analysis Phase a conversion approach was developed and it specified that data is to be brought over to the new system from an old one for historical purposes or initial load of tables, a Conversion Specification Document is needed. The conversion specification document is to map the old data to the new, specifying any logic needed to separate data currently coupled together, logic needed to create relational relationships or keys, logic to determine placement in current tables or any other process logic.

There may be occasions when developing for a new system a plan to reconstruct a data structure to feed an old system is also needed. If this is the case, a Reconstruction Specification Document is also needed.

If needed, the conversion specification program(s) are developed from these specs in the development phase.

## Development Phase Work Plan

A work plan is developed for the Development Phase listing task, estimated days to complete task and task assignment.

A work plan allows all parties to know if their perceived or real deadline is going to be met and the ability to change priorities, find additional resources or change functionality, if needed to reach a deadline.

## Phase Sign-Off

A sign-off document for the Design phase is completed for project team members, both users and developer, to sign off on. Signoff is obtained from both the assigned Information Services and responsible Bureau/Division Project Managers. Their signature indicates their confirmation that the design meets the user requirements and deliverables are acceptable per standards. If a system is being constructed utilizing parallel design and development iterations then there will be a sign off completed for each iteration completed. For major Agency project Bureau/Division Managers signatures are required after formal presentation and acceptance of the design are made. Signoff also confirms all parties agreement to the final Scope of work to which adherence will be monitored, deviations to the Scope at this point will need to be agreed to by management and project schedule adjusted accordingly.

--

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	4
		Page	1
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Development	Revised	12/20/00
<p><b>DEVELOPMENT</b></p> <p><b>Produce Iterative Development for User</b> (screens, menus, reports)  <b>Produces Test Quality Assurance Plan/Test data</b>  <b>Produce Application Testing</b> (integrated, system testing)  <b>Produce Conversion Modules/Test</b>  <b>Produce Updated System/Application Documentation</b>  <b>Produce Start of User Documentation</b>  <b>Produce Implementation Work Plan</b>  <b>Produce Systems Acceptance Sign-off</b></p> <p>This phase produces the parts of the system (screens, reports, batch processes, called routines, etc.) and demonstrates each to the designated users. Testing of each application component and of the application as a whole is completed. The prototyping and development of the application modules is completed and demonstrated until user sign-off.</p> <p>Common development processes include:</p> <ul style="list-style-type: none"> <li>- Development Phase Work Plan</li> <li>- Design, construction and demonstration of screens</li> <li>- Design, construction and demonstration of reports</li> <li>- Design, construction and demonstration of menus</li> <li>- Design, construction and demonstration of batch processes</li> <li>- Develop test quality assurance plan/test data</li> <li>- Testing and demonstration of module, integration and system functions</li> <li>- Testing of system recovery procedures</li> <li>- Determination of the application's requirements regarding the workstation operating environment and system files such as: config.sys, autoexec.bat, search path, attribute settings of sensitive files, and how these requirements may or may not affect random access memory (RAM) requirements of other applications executing in the same environment.</li> <li>- Module code walkthrus</li> <li>- Start of user documentation development</li> <li>- Application/System support documentation updates</li> <li>- Conversion program/routines are developed and tested.</li> </ul> <p>The deliverables of this phase are the actual source code, maintenance documentation, executable code, updated application/systems documentation and user documentation. (Refer to <b>Section 1.0 Standard Naming Conventions Section</b> for proper naming of modules and data items within application source. Also refer to <b>Section 4.0 Development Environment Procedural Guideline</b> on recommended procedure to follow in developing, testing or modifying application/systems per environment.)</p>			

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	4
		Page	2
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Development	Revised	12/20/00

### Source Code

Good, well written, structured source code, rich with comments is the most useful form of maintenance documentation. A standard style of coding and commenting protect our investment in code. The standards should be abbreviated enough that it does not become overly burdensome but by the same means insure against spaghetti style of coding and missed comments of vital variables, processing rules or complex logic. *(See Section 6.0 Structured Coding and Commenting by programming tool)*

### Module Specification Updating

As stated, process specifications are started in the Design Phase. As the actual construction and coding progresses through integrative prototyping the process specification documentation is also updated.

### Module Walkthru

If considered necessary by the system analyst a formal or informal walkthru is made of the module code and narrative specification to ensure Quality Control and that structural guidelines are followed. For new application developers we suggest these be done rather formally until trained in shop standards and best operating techniques. For veteran shop developers a very informal exchange of complete code review by each other is sufficient.

### Produce Test Quality Assurance Plan/Test Data

“A prototype is code that can be made to work. In contrast, a production system should be code that can hardly be made to fail.” Prototyping is a great way to develop mainline program logic. It takes planning to handle the exception situation, and a developer cannot rely on the users to figure out everything that can go wrong. Testing must not only test processing logic, it must test everything that a developer may know can go wrong that an end user might not. For instance, testing must ensure there is no way a user can get lost and confused in the system, testing must absolutely guarantee that data integrity is maintained. Following is a list of obvious problems that testing or test data must check for:

- Actions that destroy relational integrity
  - deleting records with foreign key references
  - changing key fields
  - adding records without validating foreign keys
  - adding duplicate records
- Invalid data entered
- Incorrect results of processing or computation logic

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	4
		Page	3
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Development	Revised	12/20/00

- Production workload testing to see if system performs adequately
- Locking the system by going after the same data or get inconsistent results by not locking data
- Users can access data that is suppose to be restricted, or access data they are not privileged to
- System error messages that do not give the end user a clue on how to recover
- Hanging up the system
- All user requirements for data integrity, processing and screen functionality
- All access and entry Client/Server, LAN, WAN, Internet or Intranet interfaces

Testing generally takes three levels, a Unit Test to test out one particular module, Integration Test (running data through multiple modules) and if needed, a Systems Test (running data through the whole system)

A Systems/Application Quality Assurance Test Plan is constructed. (*See Section 3.0 Documentation `System/Application Quality Assurance Test Plan Verification List`.*)

#### Application Software Testing

Following the Quality Assurance Test Plan, system and application test data and test beds are constructed that will check all necessary application functionality, user requirements, data and data base constraints, hardware, system software, and any and all interface components including manual procedures, LAN, WAN, Internet/Intranet etc.

Elements of creating a test data bed include:

- A separate user ID containing master versions of the test data with select access granted to the public
- Utility routines to copy tables from the test user ID into developers own user Ids
- A small test version of every table that will be updated by the system. Developers can easily copy a small table to their user IDs to test update program and examine results
- A small version of every table that will be reported. There should be enough data to force test reports to print about three pages
- Larger versions of test files for volume testing
- Test scripts of online update processes with data matched to the test tables
- Test data files for import process
- Test scripts for integration testing and systems test if needed.
- Test simulated production environment

The actual generation of test scripts to generate test data are performed.

# ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL

Section	2
Chapter	4
Page	4

Section Title      System Development Methodology

Issued              2/1/97

Chapter Title      Development

Revised            12/20/00

The test plan is carried out in iterative, cyclic processes, testing all conceivable conditions and all user requirements until the designated IT developer is assured of the accuracy of the code, the assigned end user(s) will verify via end user testing that the application is working with test and real world data/actions. This process is carried out for each component identified on the A Systems/Application Quality Assurance Test Plan until final verification. Upon final verification both the IT QA and Bureau/Division assigned QA tester are to sign-off on the module or Entity via the test plan document.

## Updated Systems/Application Documentation

After user testing is completed and before implementation, all systems and application documentation is to be updated if necessary. Iteration of updating all documentation components after each phase will aid in documentation being kept up-to-date. This updating process includes the completion of any recommended additional comments to be placed in source code.

## User Documentation

User documentation is a collaborative effort between end-user, designer and developer. As much on-line documentation as is practical should be employed. On-line help text, if available, should be utilized. Error messages should be as understandable as practical. All error messages and problem resolutions must be documented in the user manual and if possible, on-line access to help text that provides the problem resolution or tells where to go to find it. Screen functioning documentation is developed so a user knows the purpose and unique functions of the screen. It is essential that it include how to respond to any and all prompts.

The end user should test the instructions and documentation to ensure the processes behave in the manner indicated but also to rework documentation into normal nomenclature if necessary. User sign off should be obtained on all user documentation.

## Implementation Phase Work Plan

An implementation task plan is developed and task assignments made for the installation and final products of the project to be completed. A formal IT and Bureau/Division Management approval document is initiated and signed by all affected parties indicating that all components needed to comprise the development phase have successfully meet their requirements. (See **Section 5.0 Standard Forms** for standard sign-off form that is to be completed)

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	5
		Page	1
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Implementation	Revised	12/20/00
<p><b>IMPLEMENTATION</b></p> <p><b>Install Application</b>  <b>Implement Conversion</b> (if needed)  <b>Conduct User Training</b>  <b>Conduct Implementation Test</b>  <b>Produce Finalized User Manuals/Application Documentation</b>  <b>Conduct Implementation Phase and QA Sign-off</b></p> <p>This phase presents the fully operating production system and provides a complete guide, including training, to the users.</p> <p>Standard implementation process includes:</p> <ul style="list-style-type: none"> <li>- Finalizing User/Systems documentation which includes; <ul style="list-style-type: none"> <li>. inventory of menus, screens, reports, and files</li> <li>. diagram(s) of all relationships among databases, identifying when each is active</li> <li>. narrative of all major processing sequences. Step-by-step guide to special devices (digitizers, plotters, scanners, etc.)</li> <li>. step-by-step guide to use any special software (mail merge, Internet, etc.)</li> <li>. description of all process-initiating software (BAT files) and guide to usage</li> <li>. inventory of directories &amp; subdirectories including their hierarchic structure</li> <li>. complete list of application generated messages, meanings, and suggested actions</li> </ul> </li> <li>- Installation into production by DBA or LAN administrator via guidelines or scripts developed by developer</li> <li>- Conversion of data and initial load</li> <li>- Tune database or application as needed</li> <li>- User certification that loaded data is acceptable for production processing</li> <li>- Mapping standards of key usage</li> <li>- User training</li> <li>- Implementation test of production system</li> <li>- Finalized documentation</li> </ul> <p>Key deliverable of the Implementation Phase are:</p> <p>Trained users and Operations staff  Installed system  Converted data/loaded tables  User and System Manuals documentation  User sign-off</p>			



<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	5
		Page	2
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Implementation	Revised	12/20/00

### Installation of System

The requesting development team will complete the Client/Server System Procedures Implementation signoff document, attaching the Quality Assurance Validation Test/Approval Check list that has the IS and Bureau/Division QA testers signatures for the modules being installed. The Implementation documents are then given to the assigned production DBA or implementer to install. Upon Installation the DBA/production installer is to date and sign the Implementation signoff document and return to the responsible project team for filing in the systems documentation.

### Implementation Sign-Off

After Implementation the Developer will update the entity/attribute and module specification to include the Date and the initials of the assigned responsible IS and Bureau/Division QA verifier as reflected on the Systems/Application Quality Assurance Test Plan. This will serve as an auditable trail for new development and future enhancements and maintenance's.

An official sign-off is to take place on the completion of implementation test and before components are moved to a production environment. (*Section 5.0 Standard Forms* contains the `Sample Sign Off Documents` to be used for these purposes.) Sign off documents are to be kept with the system and application documentation manuals.

<b>ENVIRONMENTAL PROTECTION AGENCY DATA PROCESSING STANDARDS MANUAL</b>		Section	2
		Chapter	6
		Page	1
Section Title	System Development Methodology	Issued	2/1/97
Chapter Title	Post Implementation/System Shakedown	Revised	12/20/00
<p align="center"><b>POST IMPLEMENTATION/SYSTEM SHAKEDOWN</b></p> <p><b>Activate Change Control</b>  <b>Activate Development Environment Procedures</b>  <b>Activate Backup/Recovery Procedures</b>  <b>Move Documentation to Production Libraries</b></p> <p>This phase stabilizes the implemented system. In the event of hardware or other failure, the current version of the system, its most recent backup of the data, and the system's operating environment comprise the domain of recovery for the system. These components are protected by a Change Control process that logs modifications to any of them. The importance of the Change Control procedure cannot be overstated. Change Control assures that system recovery components will work and in the case of malfunction backout of all affected components can be made. It also ensures that a project is completed and does not run into endless development or non-production status.</p> <p>Phase processes includes:</p> <ul style="list-style-type: none"> <li>- implemented version control of all application components of the system</li> <li>- updating of all components/documentation due to error correction</li> <li>- move all documentation to staging libraries and have librarian move to production libraries</li> <li>- PIF enactment of any enhancements, deferred requirements or change requirements to the system</li> <li>- ongoing support for the system</li> <li>- version control for the software and its operating environment</li> </ul> <p>Phase deliverables include:</p> <ul style="list-style-type: none"> <li>- production system that runs interactively without error and on schedule</li> <li>- version control</li> <li>- routine backups</li> <li>- designated support personnel</li> <li>- documentation in production libraries</li> </ul> <p>(Reference <i>Section 4.0 Development Environment Procedural Guideline</i> for minimum standard procedure and practices to be followed for any development effort be it changes, enhancements, or new application module development)</p>			

